



DESARROLLO WEB

# INTRODUCCIÓN A LOS FRAMEWORKS DE JAVASCRIPT

# INTRODUCCIÓN A LOS FRAMEWORKS DE JAVASCRIPT: DIFERENCIAS ENTRE REACT, ANGULAR Y VUE.JS



Los marcos de trabajo de JavaScript contemporáneos han emergido como respuesta a la creciente complejidad de las aplicaciones web modernas, cada uno proporcionando soluciones distintivas para desafíos comunes como gestión de estado, composición de componentes, enrutamiento, y optimización de rendimiento. React, desarrollado por Facebook, se posiciona como una biblioteca enfocada en la construcción de interfaces de usuario a través de

un modelo declarativo que enfatiza la reutilización de componentes y flujo de datos unidireccional. Angular, desarrollado por Google, constituye un marco de trabajo completo que proporciona una arquitectura determinada basada en TypeScript, inyección de dependencias, y patrones empresariales establecidos que facilitan el desarrollo de aplicaciones de gran escala. Vue.js representa una aproximación progresiva que combina la simplicidad conceptual de React con capacidades más comprehensivas, proporcionando una curva de aprendizaje suave mientras mantiene poder para aplicaciones complejas.

Las diferencias arquitecturales entre estos marcos de trabajo reflejan filosofías distintas sobre cómo estructurar aplicaciones frontend, con implicaciones significativas para colaboración de equipo, mantenimiento de proyecto, y escalabilidad (Vara Mesa et al., 2015). La arquitectura basada en componentes de React con JSX promueve composición sobre herencia, permitiendo a desarrolladores construir UI complejas desde piezas simples y reutilizables mientras mantiene separación clara de responsabilidades. La estructura determinada de Angular con decoradores, servicios, e inyección de dependencias proporciona consistencia a través de equipos grandes, pero requiere inversión inicial más profunda en aprender patrones específicos del marco de trabajo. Vue.js ofrece componentes basados en plantillas que se sienten familiares a desarrolladores con experiencia en HTML y CSS mientras proporciona características avanzadas como enlace de datos reactivo y propiedades computadas que rivalizan el poder de marcos de trabajo más complejos.

En organizaciones de software empresarial, la selección entre estos marcos de trabajo a menudo depende de factores como experiencia existente del equipo, requisitos del proyecto, consideraciones de mantenimiento a largo plazo, y necesidades de integración con sistemas existentes. Por ejemplo, una empresa emergente desarrollando una plataforma de comercio electrónico orientada al cliente podría elegir React por su flexibilidad y capacidades de desarrollo rápido, mientras que una institución financiera grande construyendo sistemas internos de comercio podría preferir Angular por sus herramientas comprehensivas y patrones establecidos que aseguran consistencia a través de múltiples equipos de desarrollo. Vue.js a menudo sirve como una excelente opción para equipos que hacen transición desde aplicaciones basadas en jQuery o cuando la mejora progresiva es una prioridad.

Ejercicio práctico. Crear una implementación comparativa del mismo componente (lista de tareas pendientes) en React, Angular, y Vue.js para evaluar diferencias en sintaxis, enfoque, y experiencia de desarrollo.

- **Primero**, implementar en React utilizando componentes funcionales con hooks, demostrando composición de componentes, gestión de estado con useState, y manejo de eventos con eventos sintéticos.
- **Segundo**, implementar en Angular utilizando decoradores de componentes, sintaxis de plantilla con directivas estructurales como \*ngFor y \*ngIf, y enlace de datos bidireccional con [(ngModel)].
- **Tercero**, implementar en Vue.js utilizando componentes de archivo único con secciones de plantilla, script, y estilo, demostrando propiedades de datos reactivos y valores computados.
- **Cuarto**, comparar aspectos como tamaño de paquete, herramientas de desarrollo, curva de aprendizaje, y características de rendimiento de cada implementación.

El resultado es una comparación práctica que ilustra las compensaciones prácticas entre marcos de trabajo y proporciona perspicacia sobre cuál podría ser más apropiado para diferentes contextos de proyecto.